

Interview Summary	Application No. 09/900,111	Applicant(s) WILTAMUTH ET AL.	
	Examiner CHAMELI C. DAS	Art Unit 2192	

All participants (applicant, applicant's representative, PTO personnel):

- (1) CHAMELI C. DAS, PTO personnel. (3) _____
 (2) Mr. J. J. Baunach, applicant's representative. (4) _____

Date of Interview: 20 September 2005.

Type: a) ☒ Telephonic b) ☐ Video Conference
 c) ☐ Personal [copy given to: 1) ☐ applicant 2) ☐ applicant's representative]

Exhibit shown or demonstration conducted: d) ☐ Yes e) ☒ No.
 If Yes, brief description: _____

Claim(s) discussed: 1-60.

Identification of prior art discussed: _____

Agreement with respect to the claims f) ☒ was reached. g) ☐ was not reached. h) ☐ N/A.

Substance of Interview including description of the general nature of what was agreed to if an agreement was reached, or any other comments: See Continuation Sheet.

(A fuller description, if necessary, and a copy of the amendments which the examiner agreed would render the claims allowable, if available, must be attached. Also, where no copy of the amendments that would render the claims allowable is available, a summary thereof must be attached.)

THE FORMAL WRITTEN REPLY TO THE LAST OFFICE ACTION MUST INCLUDE THE SUBSTANCE OF THE INTERVIEW. (See MPEP Section 713.04). If a reply to the last Office action has already been filed, APPLICANT IS GIVEN ONE MONTH FROM THIS INTERVIEW DATE, OR THE MAILING DATE OF THIS INTERVIEW SUMMARY FORM, WHICHEVER IS LATER, TO FILE A STATEMENT OF THE SUBSTANCE OF THE INTERVIEW. See Summary of Record of Interview requirements on reverse side or on attached sheet.

Chameli C. Das
CHAMELI C. DAS
PRIMARY EXAMINER

9/24/05

Examiner Note: You must sign this form unless it is an Attachment to a signed Office action.

 Examiner's signature, if required

Summary of Record of Interview Requirements

Manual of Patent Examining Procedure (MPEP), Section 713.04, Substance of Interview Must be Made of Record

A complete written statement as to the substance of any face-to-face, video conference, or telephone interview with regard to an application must be made of record in the application whether or not an agreement with the examiner was reached at the interview.

Title 37 Code of Federal Regulations (CFR) § 1.133 Interviews

Paragraph (b)

In every instance where reconsideration is requested in view of an interview with an examiner, a complete written statement of the reasons presented at the interview as warranting favorable action must be filed by the applicant. An interview does not remove the necessity for reply to Office action as specified in §§ 1.111, 1.135. (35 U.S.C. 132)

37 CFR §1.2 Business to be transacted in writing.

All business with the Patent or Trademark Office should be transacted in writing. The personal attendance of applicants or their attorneys or agents at the Patent and Trademark Office is unnecessary. The action of the Patent and Trademark Office will be based exclusively on the written record in the Office. No attention will be paid to any alleged oral promise, stipulation, or understanding in relation to which there is disagreement or doubt.

The action of the Patent and Trademark Office cannot be based exclusively on the written record in the Office if that record is itself incomplete through the failure to record the substance of interviews.

It is the responsibility of the applicant or the attorney or agent to make the substance of an interview of record in the application file, unless the examiner indicates he or she will do so. It is the examiner's responsibility to see that such a record is made and to correct material inaccuracies which bear directly on the question of patentability.

Examiners must complete an Interview Summary Form for each interview held where a matter of substance has been discussed during the interview by checking the appropriate boxes and filling in the blanks. Discussions regarding only procedural matters, directed solely to restriction requirements for which interview recordation is otherwise provided for in Section 812.01 of the Manual of Patent Examining Procedure, or pointing out typographical errors or unreadable script in Office actions or the like, are excluded from the interview recordation procedures below. Where the substance of an interview is completely recorded in an Examiners Amendment, no separate Interview Summary Record is required.

The Interview Summary Form shall be given an appropriate Paper No., placed in the right hand portion of the file, and listed on the "Contents" section of the file wrapper. In a personal interview, a duplicate of the Form is given to the applicant (or attorney or agent) at the conclusion of the interview. In the case of a telephone or video-conference interview, the copy is mailed to the applicant's correspondence address either with or prior to the next official communication. If additional correspondence from the examiner is not likely before an allowance or if other circumstances dictate, the Form should be mailed promptly after the interview rather than with the next official communication.

The Form provides for recordation of the following information:

- Application Number (Series Code and Serial Number)
- Name of applicant
- Name of examiner
- Date of interview
- Type of interview (telephonic, video-conference, or personal)
- Name of participant(s) (applicant, attorney or agent, examiner, other PTO personnel, etc.)
- An indication whether or not an exhibit was shown or a demonstration conducted
- An identification of the specific prior art discussed
- An indication whether an agreement was reached and if so, a description of the general nature of the agreement (may be by attachment of a copy of amendments or claims agreed as being allowable). Note: Agreement as to allowability is tentative and does not restrict further action by the examiner to the contrary.
- The signature of the examiner who conducted the interview (if Form is not an attachment to a signed Office action)

It is desirable that the examiner orally remind the applicant of his or her obligation to record the substance of the interview of each case. It should be noted, however, that the Interview Summary Form will not normally be considered a complete and proper recordation of the interview unless it includes, or is supplemented by the applicant or the examiner to include, all of the applicable items required below concerning the substance of the interview.

A complete and proper recordation of the substance of any interview should include at least the following applicable items:

- 1) A brief description of the nature of any exhibit shown or any demonstration conducted,
- 2) an identification of the claims discussed,
- 3) an identification of the specific prior art discussed,
- 4) an identification of the principal proposed amendments of a substantive nature discussed, unless these are already described on the Interview Summary Form completed by the Examiner,
- 5) a brief identification of the general thrust of the principal arguments presented to the examiner,
(The identification of arguments need not be lengthy or elaborate. A verbatim or highly detailed description of the arguments is not required. The identification of the arguments is sufficient if the general nature or thrust of the principal arguments made to the examiner can be understood in the context of the application file. Of course, the applicant may desire to emphasize and fully describe those arguments which he or she feels were or might be persuasive to the examiner.)
- 6) a general indication of any other pertinent matters discussed, and
- 7) if appropriate, the general results or outcome of the interview unless already described in the Interview Summary Form completed by the examiner.

Examiners are expected to carefully review the applicant's record of the substance of an interview. If the record is not complete and accurate, the examiner will give the applicant an extendable one month time period to correct the record.

Examiner to Check for Accuracy

If the claims are allowable for other reasons of record, the examiner should send a letter setting forth the examiner's version of the statement attributed to him or her. If the record is complete and accurate, the examiner should place the indication, "Interview Record OK" on the paper recording the substance of the interview along with the date and the examiner's initials.

Continuation of Substance of Interview including description of the general nature of what was agreed to if an agreement was reached, or any other comments:

The Examiner indicated that this application would be in condition for allowance if the independent claims are amended to include the limitations of claims 12-13 (wherein said specifying of programmer intent includes providing a versioning-aware overload resolution method to locate a second method invoked by a first method invocation, wherein said versioning-aware overload resolution method includes: determining a type indicated by the first method invocation, checking up an inheritance chain until at least one applicable, accessible, non-override method declaration is found; performing overload resolution on a set of applicable, accessible, non-override methods declared for the type; and selecting the second method based on a performance of said overload resolution). The Examiner also indicated that claims 23 and 42 should be amended to overcome non-statutory rejection and claims 6, 28, and 47 should be amended to overcome 112 rejection. The applicant agreed to amend the claims as indicated by the Examiner.

Chameli C. Das
CHAMELI C. DAS
PRIMARY EXAMINER

9/21/05



INTELLECTUAL PROPERTY LAW

FACSIMILE

PHILADELPHIA

One Liberty Place, 46th Floor
Philadelphia, PA 19103
215-568-3100
Fax: 215-568-3439

SEATTLE

999 Third Avenue, Suite 1606
Seattle, WA 98104
206-332-1380
Fax: 206-624-7317

DATE: September 20, 2005

OFFICIAL PAPER

Please deliver this and the following pages to:

Examiner: **Chameli Das**

U.S.P.T.O. Group Art Unit: **2192**

Telecopier No.: **571-273-3696**

U.S. Serial No.: **09/900,111**

Client/Matter No.: **MSFT-0572**

Sender's Name: **Christina Laigo on behalf of Jeremiah J. Baunach**

Pages to Follow: **9**

If transmission is not complete, please call our Seattle Office at (206) 332-1380.

COVER MESSAGE:

OFFICIAL FACSIMILE. PLEASE DELIVER TO EXAMINER IMMEDIATELY.

Attached hereto is the following document:

- 1) Claims as amended

THIS MESSAGE IS INTENDED ONLY FOR THE USE OF THE INDIVIDUAL OR ENTITY TO WHICH IT IS ADDRESSED AND MAY CONTAIN INFORMATION THAT IS PRIVILEGED, CONFIDENTIAL AND EXEMPT FROM DISCLOSURE UNDER APPLICABLE LAW. IF THE READER OF THIS MESSAGE IS NOT THE INTENDED RECIPIENT, OR THE EMPLOYEE OR AGENT RESPONSIBLE FOR DELIVERY OF THE MESSAGE TO THE INTENDED RECIPIENT, YOU ARE HEREBY NOTIFIED THAT ANY DISSEMINATION, DISTRIBUTION OR COPYING OF THIS COMMUNICATION IS STRICTLY PROHIBITED. IF YOU HAVE RECEIVED THIS COMMUNICATION IN ERROR, PLEASE NOTIFY US IMMEDIATELY BY TELEPHONE AND RETURN THE ORIGINAL TO US AT THE ABOVE ADDRESS VIA THE U.S. POSTAL SERVICE. THANK YOU.

WOODCOCK WASHBURN LLP

A Partnership Including Professional Corporations

www.woodcock.com

JOB CODE: 194856

Proposed Amendment
Unofficial

Abi-Da
No-Cur
No-Fu
With-oh.

DOCKET NO.: MSFT-0572 / 160077.01

U.S. PATENT APP. NO. 09/900,111

1. (currently amended) A method for providing versioning support for at least one software component of an object-oriented programming language, the method operating on a computer and comprising:

specifying programmer intent with regard to versioning of said at least one software component by assigning at least one keyword to said at least one software component, wherein said specifying of programmer intent includes providing a versioning-aware overload resolution method to locate a second method invoked by a first method invocation, wherein said versioning-aware overload resolution method includes:

determining a type indicated by the first method invocation,
checking up an inheritance chain until at least one applicable,
accessible, non-override method declaration is found;

performing overload resolution on a set of applicable, accessible, non-override methods declared for the type; and

selecting the second method based on a performance of said overload resolution.

2. (original) A method according to claim 1, wherein said assigning said at least one keyword includes assigning at least one of virtual, new and override keywords.

3. (original) A method according to claim 1, wherein said assigning said at least one keyword to said at least one software component specifies programmer intent with regard to whether said at least one software component overrides another software component.

4. (original) A method according to claim 1, wherein said assigning said at least one keyword to said at least one software component specifies programmer intent with regard to whether said at least one software component is capable of being overridden by another software component.

5. (original) A method according to claim 1, wherein said assigning said at least one keyword to said at least one software component specifies programmer intent with regard to whether said at least one software component hides another software component.

- 2 -

DOCKET NO.: MSFT-0572 / 160077.01

U.S. PATENT APP. NO. 09/900,111

6. (currently amended) A method according to claim 1, wherein said at least one software component is at least one member of the object-oriented programming language and the object-oriented programming language is from one of the sources of origin identified by C#, FORTRAN, PASCAL, VISUAL BASIC, C, C++, and JAVA ~~Fortran, Pascal, Visual Basic, C, C++ and Java.~~

7. (original) A method according to claim 1, wherein said specifying of programmer intent includes assigning intelligent defaults to said at least one software component in the absence of assigning said at least one keyword to said at least one software component.

8. (original) A method according to claim 7, wherein when programmer intent is not fully specified, the compiler of the programming language produces a warning before assigning said intelligent defaults.

9. (currently amended) A method according to claim 7, wherein said assigning of intelligent defaults includes assigning to said at least one software component the most limited form of accessibility, based upon the a type of said at least one software component.

10. (original) A method according to claim 7, wherein by default, when said at least one software component is at least one method declaration with no accessibility modifiers appearing in the corresponding class, the at least one method declaration is defaulted to be private to that class.

11. (original) A method according to claim 7, wherein by default, said at least one software component is non-virtual, rather than virtual.

12. (canceled).

13. (canceled).

14. (currently amended) A method according to claim ~~12~~ 1, wherein for a virtual method invocation, said versioning-aware overload resolution method includes determining

- 3 -

DOCKET NO.: MSFT-0572 / 160077.01

U.S. PATENT APP. NO. 09/900,111

the second method based on ~~a the~~ run-time type of ~~an the~~ instance of the first method invocation.

15. (currently amended) A method according to claim ~~12~~ 1, wherein for a non-virtual method invocation, said versioning-aware overload resolution method includes determining the second method based on ~~a the~~ compile-time type of ~~a the~~ instance of the first method invocation.

16. ((currently amended) A method according to claim ~~12~~ 1, wherein said versioning-aware overload resolution method includes bounding names at run-time, and not bounding offsets at compile-time.

✓ 17. (currently amended) A method according to claim ~~12~~ 1, wherein the overload resolution method prevents a base software component from breaking ~~a the~~ functionality of a derived software component when versioning the base software component and such breaking is not intended by the programmer.

18. (original) A method according to claim 1, wherein said at least one software component is binary compatible with code utilizing other versions of said at least one software component.

19. (original) A method according to claim 1, wherein said at least one software component is source compatible with code utilizing other versions of said at least one software component.

20. (original) A computer readable medium bearing computer executable instructions for carrying out the method of claim 1.

21. (original) A modulated data signal carrying computer executable instructions for performing the method of claim 1.

22. (original) A computing device comprising means for performing the method of claim 1.

- 4 -

DOCKET NO.: MSFT-0572 / 160077.01

U.S. PATENT APP. NO. 09/900,111

23. (currently amended) A computer readable storage medium having stored thereon a plurality of computer-executable modules written in an object-oriented programming language, the computer executable modules comprising:

a versioning mechanism enabling a programmer to specify intent with regard to versioning of at least one software component by assigning at least one keyword to said at least one software component, wherein said specifying of programmer intent includes providing a versioning-aware overload resolution method to locate a second method invoked by a first method invocation, wherein said versioning-aware overload resolution method includes:

determining a type indicated by the first method invocation,
checking up an inheritance chain until at least one applicable,
accessible, non-override method declaration is found;
performing overload resolution on a set of applicable, accessible, non-
override methods declared for the type; and
selecting the second method based on a performance of said overload
resolution.

24. (original) A computer readable medium according to claim 23, wherein said assigning said at least one keyword includes assigning at least one of virtual, new and override keywords.

25. (original) A computer readable medium according to claim 23, wherein said assigning said at least one keyword to said at least one software component specifies programmer intent with regard to whether said at least one software component overrides another software component.

26. (original) A computer readable medium according to claim 23, wherein said assigning said at least one keyword to said at least one software component specifies programmer intent with regard to whether said at least one software component is capable of being overridden by another software component.

- 5 -

DOCKET NO.: MSFT-0572 / 160077.01

U.S. PATENT APP. NO. 09/900,111

27. (original) A computer readable medium according to claim 23, wherein said assigning said at least one keyword to said at least one software component specifies programmer intent with regard to whether said at least one software component hides another software component.

28. (currently amended) A computer readable medium according to claim 23, wherein said at least one software component is at least one member of the object-oriented programming language and the object-oriented programming language is from one of the sources of origin identified by C#, FORTRAN, PASCAL, VISUAL BASIC, C, C++, and JAVA ~~Fortran, Pascal, Visual Basic, C, C++ and Java.~~

29. (original) A computer readable medium according to claim 23, wherein said specifying of programmer intent includes assigning intelligent defaults to said at least one software component in the absence of assigning said at least one keyword to said at least one software component.

30. (original) A computer readable medium according to claim 29, wherein when programmer intent is not fully specified, the compiler of the programming language produces a warning before assigning said intelligent defaults.

31. (currently amended) A computer readable medium according to claim 29, wherein said assigning of intelligent defaults includes assigning to said at least one software component the most limited form of accessibility, based upon the a type of said at least one software component.

32. (original) A computer readable medium according to claim 29, wherein by default, when said at least one software component is at least one method declaration with no accessibility modifiers appearing in the corresponding class, the at least one method declaration is defaulted to be private to that class.

33. (original) A computer readable medium according to claim 29, wherein by default, said at least one software component is non-virtual, rather than virtual.

34. (canceled)

- 6 -

DOCKET NO.: MSFT-0572 / 160077.01

U.S. PATENT APP. NO. 09/900,111

35. (canceled)

36. (currently amended) A computer readable medium according to claim 34- 23, wherein for a virtual method invocation, said versioning-aware overload resolution method includes determining the second method based on a the run-time type of an the instance of the first method invocation.

37. (currently amended) A computer readable medium according to claim 34- 23, wherein for a non-virtual method invocation, said versioning-aware overload resolution method includes determining the second method based on a the compile-time type of an the instance of the first method invocation.

✓ 38. (currently amended) A computer readable medium according to claim 34- 23, ✓ wherein said versioning-aware overload resolution method includes bounding names at run-time, and not bounding offsets at compile-time.

✓ 39. (currently amended) A computer readable medium according to claim 34- 23, wherein the overload resolution method prevents a base software component from breaking a the functionality of a derived software component when versioning the base software component and such breaking is not intended by the programmer.

40. (original) A computer readable medium according to claim 23, wherein said at least one software component is binary compatible with code utilizing other versions of said at least one software component.

41. (original) A computer readable medium according to claim 23, wherein said at least one software component is source compatible with code utilizing other versions of said at least one software component.

42. (currently amended) An object-oriented programming language stored on a computer for producing computer executable modules, comprising:

a versioning mechanism enabling a programmer to specify intent with regard to versioning of at least one software component by assigning at least one keyword to said at least one software component, wherein said specifying of programmer intent includes

- 7 -

DOCKET NO.: MSFT-0572 / 160077.01

U.S. PATENT APP. NO. 09/900,111

providing a versioning-aware overload resolution method to locate a second method invoked by a first method invocation, wherein said versioning-aware overload resolution method includes:

determining a type indicated by the first method invocation,
checking up an inheritance chain until at least one applicable,
accessible, non-override method declaration is found;
performing overload resolution on a set of applicable, accessible, non-
override methods declared for the type; and
selecting the second method based on a performance of said overload
resolution.

43. (currently amended) An object-oriented programming language stored on a computer according to claim 42, wherein said assigning said at least one keyword includes assigning at least one of virtual, new and override keywords.

44. (currently amended) An object-oriented programming language stored on a computer according to claim 42, wherein said assigning said at least one keyword to said at least one software component specifies programmer intent with regard to whether said at least one software component overrides another software component.

45. (currently amended) An object-oriented programming language stored on a computer according to claim 42, wherein said assigning said at least one keyword to said at least one software component specifies programmer intent with regard to whether said at least one software component is capable of being overridden by another software component.

46. (currently amended) An object-oriented programming language stored on a computer according to claim 42, wherein said assigning said at least one keyword to said at least one software component specifies programmer intent with regard to whether said at least one software component hides another software component.

47. (currently amended) An object-oriented programming language stored on a computer according to claim 42, wherein said at least one software component is at least one member of the object-oriented programming language and the object-oriented programming

- 8 -

DOCKET NO.: MSFT-0572 / 160077.01

U.S. PATENT APP. NO. 09/900,111

language is from one of the sources of origin identified by C#, FORTRAN, PASCAL, VISUAL BASIC, C, C++, and JAVA ~~Fortran, Pascal, Visual Basic, C, C++ and Java.~~

48. (currently amended) An object-oriented programming language stored on a computer according to claim 42, wherein said specifying of programmer intent includes assigning intelligent defaults to said at least one software component in the absence of assigning said at least one keyword to said at least one software component.

49. (currently amended) An object-oriented programming language stored on a computer according to claim 48, wherein when programmer intent is not fully specified, the compiler of the programming language produces a warning before assigning said intelligent defaults.

50. (currently amended) An object-oriented programming language stored on a computer according to claim 48, wherein said assigning of intelligent defaults includes assigning to said at least one software component the most limited form of accessibility, based upon ~~the~~ a type of said at least one software component.

51. (currently amended) An object-oriented programming language stored on a computer according to claim 48, wherein by default, when said at least one software component is at least one method declaration with no accessibility modifiers appearing in the corresponding class, the at least one method declaration is defaulted to be private to that class.

52. (currently amended) An object-oriented programming language stored on a computer according to claim 48, wherein by default, said at least one software component is non-virtual, rather than virtual.

53. (canceled)

54. (canceled)

55. (currently amended) An object-oriented programming language stored on a computer according to claim ~~53~~ 42, wherein for a virtual method invocation, said versioning-

- 9 -

DOCKET NO.: MSFT-0572 / 160077.01

U.S. PATENT APP. NO. 09/900,111

aware overload resolution method includes determining the second method based on a the run-time type of an the instance of the first method invocation.

56. (currently amended) An object-oriented programming language stored on a computer according to claim ~~53~~ 42, wherein for a non-virtual method invocation, said versioning-aware overload resolution method includes determining the second method based on a the compile-time type of the instance of the first method invocation.

57. (currently amended) An object-oriented programming language stored on a computer according to claim ~~53~~ 42, wherein said versioning-aware overload resolution method includes bounding names at run-time, and not bounding offsets at compile-time.

58. (currently amended) An object-oriented programming language stored on a computer according to claim ~~53~~ 42, wherein the overload resolution method prevents a base software component from breaking a the functionality of a derived software component when versioning the base software component and such breaking is not intended by the programmer.

59. (currently amended) An object-oriented programming language stored on a computer according to claim 42, wherein said at least one software component is binary compatible with code utilizing other versions of said at least one software component.

60. (currently amended) An object-oriented programming language stored on a computer according to claim 42, wherein said at least one software component is source compatible with code utilizing other versions of said at least one software component.